Amendments to the Specification:

Please replace the paragraph beginning on page 5, line 8, with the following amended paragraph.

To achieve the above object, the method of this invention for synchronous serial communication comprising the step of transmitting data serially through data lines in synchrony with timing signals sent through clock line, wherein, if the transmission of a specified length of serial data is required, transmission of the data is achieved by dividing the data for transmission into a plurality of blocks, firstly transmitting block information notifying the block(s) to be transmitted, and then transmitting the data included in the block(s) notified by the block information, and, on the data included in the block(s) not notified, the corresponding previous data stored in the data receiving component are used (second mode).

Please replace the paragraph beginning on page 7, line 21, with the following amended paragraph.

[Constitution] A first embodiment of this invention will be described below with reference to attached figures. Fig. 1 is a schematic diagram to indicate the outline of an image forming system based on thermal ink jet printing representing a first embodiment of this invention. The image forming system 100 based on thermal ink jet printing shown in Fig. 1 comprises an engine portion 11 including a CPU 111 and a data transmission controller 112, and a

recording head portion 12 including a data reception controller 113 and a printer head 114. Data ~~reception~~ transmission controller 112 is connected through a bus to CPU 111, and writes data in response to an instruction from CPU 111, or receives an instruction from CPU 111 for starting communication. Data transmission controller 112, on receipt of an instruction from CPU 111 for starting communication, determines which mode out of the communication modes described below will be most suitable for transmitting the data in a shortest possible time, adds the communication mode information informing the communication mode thus chosen to the data to be transmitted, and sends the resulting data through serial data communication to the data reception controller 113. This system communicates data between data transmission controller 112 and data reception controller 113 through synchronous serial communication. The data to be transmitted s111 are delivered through a data line in synchrony with clock signals s113 delivered through a clock line.

Please replace the paragraph beginning on page 11, line 17, with the following amended paragraph.

Let's assume now the data transmission controller will transmit data based on the block mode. Then, the controller will set data corresponding to a start bit to 1, and the data corresponding to 2nd and 3rd bits to 0 and 1, then set the block information for indicating the blocks to be transmitted from the highest block to the lowest block in order. The controller will send the data included in

the blocks which are set by the block information, one after another with 8 bits in order from high-order bit. With regard to the example shown in Fig. 2(b), the data are set at 1 to the 9th and 12th bits in terms of clock signals. This indicate changes are introduced in the 3rd block (data corresponding to 31st to 24th bit) and in the 0th block (data corresponding to 7th to 0th bit). At this time, since the data in the other blocks are not changed, the data in the other blocks will not be transmitted. In Fig. 2(b), a change is introduced as a 0Fhex (hexadecimal data). When the data reception controller receives a series of data as shown in Fig. ~~2(c)~~ 2(b), it predicts, by recognizing the data corresponding to the second and third bits which are sent following a start bit are 0 and 1, the data will be transmitted through the block mode. Then, the same controller recognizes the data continuing from 5th to 12th bits in terms of clock signals carry the information about the blocks transmitted. If data transmission based on the block mode occurs as shown in Fig. 2(b), it indicates changes are introduced in the blocks 3 and 0, but the other blocks remain unchanged. When data transmission occurs based on the block mode, changed data will be delivered one after another with 8 bit from 13rd bit onward in terms of clock signals. In the particular example shown in Fig. 2(b), two blocks of data amounting to 16 bits in total are delivered. Accordingly, when the two blocks of data are delivered through the block mode as shown in Fig. 2(b), the data will require, for their delivery, 28 bits or 14 clock signals. If the same amount of data were delivered through the full mode, it would require 34 clock signals as was indicated with

reference to Fig. 2(a). Therefore, data transmission based on the block mode could save a time corresponding to 20 clock signals, as compared with data transmission based on the full mode. ~~In other words, as long as argument is limited to the particular example as depicted in Figs. 2(a) and 2(b), transmitting data through the block mode will shorten the transmission time by a time corresponding to 20 clock signals, as compared with the conventional data transmission based on the full mode.~~ That is, the block information notifying the block(s) to be transmitted is transmitted first, and then the data included in the block(s) notified by the block information is transmitted, while the data included in the block(s) not notified is not transmitted, and the object of the present invention to improve the transfer speed can be achieved.

Please replace the paragraph beginning on page 14, line 11, with the following amended paragraph.

[Constitution of data transmission controller] Next, the data transmission controller will be described with reference to Fig. 4. Fig. 4 is a block diagram of the data transmission controller 112. The data transmission controller 112 comprises an address decoding portion 121 which decodes address data sent by a CPU, and determines the locations into which the data received should be written; a data register 122 which temporarily stores the data sent by the CPU in the locations nominated by the address decoding portion 121; a memory 124 for current blocks which stores currently changed blocks of data; a memory 125

for previous blocks which stores previously changed blocks of data; a block comparing/calculating portion 126 which checks by comparison whether or not the current block information are the same with the previous ones, and, if transmission of a data is required, calculates the number of clock signals required for the transmission of the data for each of the three communication modes; a decision portion 127 which decides the communication mode most suitable for the transmission of the data, based on the comparison/calculation results from the block comparing/calculating portion 126; an output circuit portion 128 which transmits the data stored in the data register portion 122 through the mode recommended by the decision portion 127; and a control portion 123 which controls the operations of the memory for current block, memory for previous block, block comparing/calculating portion, output circuit portion, etc.

Please replace the paragraph beginning on page 15, line 12, with the following amended paragraph.

[Operation of data transmission controller] CPU 111 creates a chip select signal s121, write signal s123, addresses ~~b122~~ s122 and data ~~125~~ s125 based on the data sent from external personal computers or the like and sends them to the data transmission controller 112. In the data transmission controller, the address decoding portion 121 analyzes addresses sent from CPU, and dispatches enable signals s124 according to which the locations of data register portion 122 to which data should be written are determined. The enable signal is a 8 bit bus,

and makes it possible to replace the data assigned to a certain address of the data register with a new data, by changing the bit corresponding to that address to 1, thereby nominating the address at which replacement should occur. The address decoding portion 121 obtains the information s127 notifying which block of data have been currently updated, based on the addresses ~~b123~~ s123 sent from CPU, and sends the information s127 to the memory for current blocks. In addition, address decoding portion ~~122~~ 121, on receipt of a signal for starting data transmission from CPU, sends a start trigger signal s126 to the control portion 123. Control portion 123, on receipt of the start trigger signal s126, creates an instruction signal ~~130~~ s130 and sends the signal to the block comparing/calculating portion 126 to urge it to compare the block information s128 about the blocks currently written which is stored in the memory for current blocks with the corresponding information s129 about the blocks previously written which is stored in the memory for previous blocks, and to calculate the number of clock signals required for the transmission of data to be currently transmitted for each of the three communication modes. The block comparing/calculating portion 126, on receipt of the instruction signal s130 from the control portion, compares the current block information s128 with the previous block information s129, and, when it finds the two are coincident with each other, it will send a status signal s131 to the decision portion. And the block comparing/calculating portion 126 calculates the number of clock signals required for transmitting those data on the burst mode and on the block mode,

and sends signals s132 and s133 representing the calculation results for the two modes to the decision portion.

Please replace the paragraph beginning on page 18, line 12, with the following amended paragraph.

[Operation of block comparing/calculating portion] Fig. 5 is a flowchart to indicate the processes achieved by a data block comparing/calculating portion. At step ~~10~~ S10, the block comparing/calculating portion 126 checks whether or not it has received an instruction signal s130 from control portion 126 which urges portion ~~126~~ 123 to make a comparison/calculation, and when the same portion finds it has received the instruction signal ~~130~~ s130 in question, it proceeds to step ~~11~~ S11. Here, the block comparing/calculating portion 126 compares block information s128 of the current blocks with block information s129 of the previous blocks. When the portion 126 finds the two are the same, it proceeds to step ~~12~~ S12 and makes a calculation consisting of multiplying the number of blocks by 4 and adding 2 (the number of clock signals required for the start bit and the setting of the mode) to the answer, and obtains the number of clock signals required for transmitting the data on the burst mode. At step ~~13~~ S13, the portion 126 hoists a flag s131 signifying the current data may be transmitted on the burst mode. On the other hand, if the block comparing/calculating portion 126 finds the block information s128 of current data does not coincide with the block information s129 of previous data, it

proceeds to step ~~14~~ S14, and makes a calculation consisting of multiplying the number of blocks by 4 and adding 2 (the number of clock signals required for the start bit and the setting of the mode) to the answer, and of further adding the number of clock signals required for the setting of block information, to the previous answer, to obtain thereby the number of clock signals required for transmitting the current data on the block mode. In this way, the portion 126 calculates the number of clock signals required for transmitting the current data for each one of the communication modes. However, if the current data must be sent through the full mode, the number of clock signals required for the transmission will be 34, and thus no additional calculation will be necessary, as long as the same number of data are transmitted through this mode.

Please replace the paragraph beginning on page 19, line 23, with the following amended paragraph.

[Operation of decision portion] Fig. 6 is a flowchart to indicate the processes achieved by a decision portion 127. At step ~~20~~ S20, the portion in question checks whether or not there is hoisted at block comparing/calculating portion 126 a flag s131 signifying the two blocks coincide with each other, and when it finds the flag is hoisted, it proceeds to step ~~21~~ S21. At step ~~21~~ S21, the decision portion 127 determines whether the current data transmission should be made on the mode giving the less calculation result out of values of signals s132 and s133 (or whether the current data transmission should occur on the

block mode or on the burst mode), or whether all the current data should be transmitted on the full mode to replace all the previous data stored in the data register 122. On the contrary, when the decision portion finds the flag in question is not hoisted, it proceeds to step ~~22~~ S22. At step ~~22~~ S22, decision portion 127 compares the number of clocks signals required for transmitting the data on the block mode obtained by the block comparing/calculating portion 126 and informed through signal s133, with the corresponding number required for transmitting the same data on the full mode. If decision portion 127 finds, as a result of comparison, transmission of the data on the full mode will require less clock signals, at step ~~24~~ S24, the portion 127 delivers a signal s134 ("00") signifying the data will be transmitted on the full mode, to the output circuit portion. On the contrary, if decision portion 127 finds transmission of the data on the block mode will require less clock signals, the portion proceeds to step ~~23~~ S23, and delivers a signal s134 ("01") signifying the data will be transmitted on the block mode, to the output circuit portion.

Please replace the paragraph beginning on page 21, line 17, with the following amended paragraph.

[Operation of output circuit portion] Fig. 8 is a flowchart to indicate the output flow of the output circuit portion 128. When output circuit portion 128 finds, at step ~~30~~ S30, it has received an output trigger s136 from the control portion, it decodes a mode information s134 dispatched by the decision portion to

identify the mode for data transmission, and delivers data according to the mode thus identified. Specifically, at step ~~31~~ S31, output circuit portion 128 checks whether or not the mode information s134 is "00". When it finds the information in question is "00", it proceeds to step ~~32~~ S32, and fetches data stored in the data register portion 122, and delivers those data to the data reception controller on the full mode as indicated in Fig. 2(a). Alternatively, when it finds at step ~~31~~ S31 the mode information s134 is not "00", it proceeds to step ~~33~~ S33, and checks whether or not the information s134 is "01". When it finds the answer is affirmative, it proceeds to step ~~34~~ S34, and delivers relevant data to the data reception controller on the block mode as indicated in Fig. 2(b). Alternatively, when output circuit portion 128 finds at step ~~33~~ S33 the mode information s134 is not "01", it proceeds to step ~~35~~ S35, and delivers relevant data to the data reception controller on the burst mode as indicated in Fig. 2(c).

Please replace the paragraph beginning on page 25, line 23, with the following amended paragraph.

[Operation of data reception controller] Fig. 11 is a flowchart to indicate the operation of the data reception controller. At step ~~40~~ S40, when decoding portion 150 detects the data of a start bit is "1", data reception becomes ready. At step ~~41~~ S41, the data reception controller checks whether or not the data of the second bit is "1", and, when it finds the data in question is "1", it proceeds to step ~~42~~ S42. At step ~~42~~ S42, the data reception controller handles the data

received, and performs the operation of the burst mode, that is to say, stores the data one after another in the relevant block of the data register portion, by referring to the previous block information. If at step ~~41~~ S41 the data reception controller finds the data of the second bit is not "1", it proceeds to step ~~43~~ S43, and checks whether or not the data of the third bit is "1". If the controller finds the data in question is "1", it proceeds to step ~~44~~ S44, and handles the data received, and performs the operation of the block mode, that is to say, stores, in the block memory, the block information notifying whether or not data for each block should be written into the memory, according to the signals subsequently transmitted. Then, the data reception controller stores the data corresponding to the relevant block in a descending order of block in the data register portion, by referring to the block information stored in the block memory. However, if at step ~~43~~ S43 the data reception controller finds the data of the third bit is not "1", it proceeds to step ~~46~~ S46, and handles the data received, and performs the operation of the full mode, that is to say, stores all the transmitted data in a descending order in the data register. Then, at step ~~47~~ S47, it stores the block information notifying all the data corresponding to every block have been completely written, into the block memory.

Please replace the paragraph beginning on page 31, line 11, with the following amended paragraph.

Fig. 16 illustrates the flow of processes of a serial controller serving as one of the data receiving elements. It is the same with the corresponding one of the first embodiment shown in Fig. 11 in steps ~~40~~ S40 to ~~47~~ S47, and is different only in step ~~51~~ S51 onward. Accordingly, here steps ~~40~~ S40 to ~~47~~ S47 will be omitted from description, and only step ~~51~~ S51 onward will be described. When the serial controller finds, at step ~~41~~ S41, the data of 2nd bit is 1, it proceeds to step ~~51~~ S51. Then, it checks whether or not the data of 3rd bit is 1. When the serial controller finds the data in question is not 1, it proceeds to step ~~42~~ S42, and performs similar operations as in step ~~42~~ S42 of the first embodiment. On the contrary, if the serial controller finds, at step ~~51~~ S51, the data of 3rd bit is 1, it proceeds to step ~~52~~ S52, and checks whether or not the data of 4th bit is 1. When the controller finds the data in question is not 1, it proceeds to step ~~53~~ S53, and inverts the data corresponding to the specified blocks. On the contrary, if the serial controller finds the data of 4th bit is 1, it proceeds to step ~~54~~ S54, and checks whether or not the data of 5th bit is 1. When it finds the data in question is 1, it proceeds to step ~~55~~ S55, and checks whether or not the data of 6th bit is 1. When the serial controller finds the data of 6th bit is not 1, it proceeds to step ~~56~~ S56, shifts the data corresponding to the specified blocks leftward, and sets the least significant bit to 0. On the contrary, if the serial controller finds, at step ~~55~~ S55, the data of 6th bit is 1, it proceeds to step ~~57~~ S57, shifts the data

corresponding to the specified blocks leftward, and sets the least significant bit to 1. When the controller finds, at step ~~54~~ S54, the data of 5th bit is not 1, it proceeds to step ~~60~~ S60, and checks whether or not the data of 6th bit is 1. If it finds the data in question is 1, it proceeds to step ~~61~~ S61, shits the data corresponding to the specified blocks rightward, and sets the most significant bit to 1. If it finds the data of 6th bit is not 1, it proceeds to step ~~62~~ S62, shifts the data corresponding to the specified blocks rightward, and sets the most significant bit to 0.